

# Chapter 23

## THE PATH FORWARD

---

In the first 22 chapters of this book, I have laid out the Scrum framework and explained what I believe to be essential Scrum. You should now understand the mechanics of using Scrum to deliver innovative solutions. You should also have a good sense of why Scrum prescribes particular roles, practices, artifacts, and rules. Now you are ready to define your path forward. In this chapter I discuss the idea that there is no universal, final target for your Scrum implementation; instead, you need to define your own unique route towards agility. I end by describing the role of best practices and how to use Scrum, with its iterative and incremental approach, as the basis for discovering your own path forward.

### There Is No End State

Every organization has a vision for what it wants to achieve. Using Scrum can help organizations manage the work to achieve that vision. Being highly proficient with Scrum and thus being more agile, however, is not the end goal, but rather a means of more effectively and economically achieving business goals. So, how do you know when you're done with Scrum?

The thing is, there is no definition of done for a Scrum adoption or transition effort. There is no agile maturity model like CMMI (SEI 2010), where the goal is to try to reach level 5. Trying to define “done” for your Scrum implementation presumes that when you achieve that state, you can't get any better. This flies in the face of Scrum being a form of continuous improvement, where you are always working to better align your use of Scrum to the complex world in which you must develop products.

Worse yet, if we tried to define such an end state for the industry, it would further assume that the final target state would apply to every organization, even those that develop radically different types of products under very different circumstances.

To say, “I have finally achieved agility!” is a meaningless comment. As Mike Cohn nicely summarized, “Agile is not something you become, it's something you become more of” (Cohn 2010). There is no end state that you can call agile or Scrum. Instead, becoming more proficient with Scrum and more agile is a process of continuous, never-ending improvement aimed at improving your bottom line.

## Discover Your Own Path

Just as no one can tell you where your Scrum implementation should end, no one can lead you down a predefined path that will guarantee success. Instead, you must learn, inspect, and adapt your way forward based on your organization's own unique goals and culture and the ever-changing complex environment in which you must operate. Trying to follow someone else's path and leveraging their learning might feel like the fast track to becoming agile. However, no two organizations, or even teams within an organization, are the same. Following someone else's path might well lead you to precisely the wrong location.

You can't circumvent your own learning process. Instead, you need to quickly close your own learning loops and inspect and adapt based on what you learn. I'm not suggesting that you ignore those who have preceded you down the agile path. Examine what they have done and how it worked for them, but discover your own path for becoming more agile.

## Sharing Best Practices

If we aren't supposed to follow others' paths, how do best practices fit into the picture? Just as there is no one path to follow, there is no one set of best practices across all organizations.

When I'm asked to describe "best practices" that other organizations are using to become more agile, I give examples. However, I always provide the relevant context of the other organizations so that the people who are asking can evaluate whether it would be sensible to adopt a similar approach in their organization. Even when scaling up within a single organization, we need to be cautious about universally applying best practices. Many organizations try to describe what a successful Scrum team is doing, capture it, and then institutionalize it as a best practice. Doing so can be harmful, because these are individual team approaches that may not work for other teams.

You might notice I just used the word *approach* several times when referring to best *practices*. Let's talk for a moment about that distinction. Throughout this book I have used the term *practice* to mean a core or essential aspect of Scrum. An approach is a particular implementation of a Scrum practice. When people ask me about best *practices*, I take that to mean best *approaches*.

While two different teams or organizations have unique Scrum implementations, each should adhere to the same Scrum practices. Both, for example, should have Scrum teams made up of a product owner, ScrumMaster, and development team. Both should perform sprint planning, daily scrums, sprint reviews, and sprint retrospectives. However, I expect that each team (or organization) will have its own unique approaches to performing these practices. Let me illustrate by example.

The daily scrum is a core Scrum practice. If you don't do it, you aren't doing Scrum. During the daily scrum each team member updates and synchronizes with all other team members to get a shared picture of the full scope of work. But when the daily scrum starts, which person should provide his update first? Scrum doesn't define this. Each team will have its own approach.

For example, while working with a team in Vancouver, Canada, I learned of an interesting approach to deciding who speaks first. On this team, at the start of each daily scrum, the ScrumMaster would toss a toy stuffed moose into the air. Whoever caught the moose would speak first, and then the rest of the team members would speak in turn, moving to the left of the person who caught the moose. This simple, kind-of-silly-but-fun approach worked very well for the team in Vancouver.

As it turns out, the Vancouver team had a sister team in China that was formed some months after the Vancouver team had been established. A member of the China team asked the Vancouver team for a "policy or best practice" for determining who should speak first at the daily scrum. The Vancouver team told the person in China that in Vancouver they "tossed the moose" each daily scrum to figure that out. Apparently, tossing the moose took on a whole different meaning when translated into Chinese! An approach that worked well for the Vancouver team would not work at all for the China team. The China team adopted its own approach, as well it should.

Scrum defines core practices that must be followed. It is left to each team, however, to determine the approaches (or best practices) that work best. Approaches, therefore, are unique to each team, and they can and should be reused by other teams only if they make sense in the context of those other teams.

## Using Scrum to Discover the Path Forward

Whether you are new to Scrum or are already using Scrum to develop products, you can use the principles of Scrum to help guide you on the path forward. Mike Cohn, in *Succeeding with Agile* (Cohn 2009), goes into great detail describing this approach, and I refer you to his excellent book for a detailed treatment of the topic.

I describe the essence of this approach with an example. In 2007 I was retained to train and coach a large, multinational organization on its adoption and use of Scrum. The organization had 100 IT members in New York City and 400 IT members in Mumbai, India. At any one time the IT organization had about 45 development efforts in flight.

The organization decided that any new team that was going to use Scrum should have a coach available to help it. With limited initial coaching capacity, it was unreasonable to transition the entire IT organization over to Scrum at once. So, as is typical in such environments, the organization selected a small number of pilot efforts to focus on first. The goal was to incrementally move other teams over to Scrum as the organization's Scrum coaching capacity increased via the on-the-job training of internal coaches.

These pilot efforts crossed the spectrum from simple systems maintenance to larger, new-product development. Based on this diversity, each Scrum team implemented its own version of the Scrum framework, using approaches aligned with the people on the team and the work they had to perform. A wiki was used to “syndicate” the approaches used by each team to assist with overall organizational learning and share what was working for other teams.

Several months into the adoption, it was time to scale the use of Scrum from the team level to the organizational level. At that point we created what Cohn calls an ETC or Enterprise Transition Community (Cohn 2009). In our case the ETC was called the Working Software Group. That group of managers and executives maintained a backlog of improvement-related items and ran three-week sprints against that backlog. The items in that backlog represented organizational change initiatives (such as “Update the compensation model to be more team focused”) or significant impediments that were blocking one or more Scrum teams (“Improve server stability so that teams can complete their testing”).

By working in sprints against this improvement backlog, the organization was able to iteratively and incrementally make progress down its own path to successfully adopting Scrum. There was no predetermined end state for the organization’s use of Scrum. Trying to create one up front would have been as wasteful as trying to create a full, complete requirements specification for a totally new, never-been-built product that no one understood well.

Instead, the Working Software Group took input from the Scrum teams and stakeholders and made incremental improvements to the organizational structure to better align itself with agile values. Through continuous learning, inspecting, and adapting, the organization determined an appropriate path forward that was aligned with the overall organizational business goals.

This ETC-type pattern is quite common today. Many organizations realize that using Scrum to adopt Scrum is a sensible approach to iteratively and incrementally becoming more agile.

## Get Going!

I am frequently amused when the same people who believe that it is not possible to get the requirements for a product correct up front, and therefore want to use Scrum for development, will in the next breath explain how they aren’t quite ready to start using Scrum because they haven’t worked out all of the details of their Scrum approach! This type of thinking is poorly aligned with fundamental Scrum principles.

When employing Scrum, you shouldn’t worry about getting things perfect up front. You can’t! And trying to be perfect up front will force you to guess at the expense of important learning that you will get only by applying Scrum and seeing what happens. In my experience, most teams’ first couple of sprints aren’t all that

pretty. That's OK. My only expectation of any Scrum team is that it be better in the next sprint than it was in the previous sprint. So, don't delay getting started. Whatever you think you know now about your use of Scrum, imagine how much more you will really know after you start and finish the next sprint!

Also, don't expect your Scrum adoption to be problem free. I can guarantee that at some point your organization will encounter impediments that make performing Scrum difficult. Scrum makes visible the dysfunctions and waste that prevent organizations from reaching their true potential. What it does not do is tell organizations how to solve those issues. That hard work is up to the people in the organization.

The status quo is a powerful force. It is frequently easier for people to ignore Scrum or change Scrum than it is to change long-held organizational processes, rules, or behaviors. And a culture that is outright hostile toward being shown its dysfunctions will quickly extinguish the bright light that is exposing what lurks in the shadows. To counteract this tendency, be a steadfast and patient force for change in your organization. Understand that resistance to change is natural. Help overcome the worst of it by educating others about the principles that underlie Scrum and the goals you are trying to achieve. Work with them, rather than against them, to chip away at the obstacles that prevent your team, your product development effort, and your organization from realizing the full benefits of your Scrum implementation.

My hope is that this book has provided you with the essential Scrum knowledge to shine a bright light that will illuminate your path forward. I wish you the best of success on your journey with Scrum.